COMSC-210 Lab 4
A Dynamic Indexed Array Template

LAB 4a: Write A Dynamic Array Class Template `[ DynamicArray.h and DynamicArrayDriver.cpp ]`
Purpose. The purpose of this lab is for you to write a templated class using dynamic memory. It's exactly the same as the **StaticArray** template from lab 3a, except that the data array is dynamically allocated and expandable when a higher index is requested in the operator[ ] setter (as explained in the lecture notes).

Requirements. Write **`DynamicArray.h`** per our lecture topic 3 and 4 notes, and **`DynamicArrayDriver.cpp`** to test it. Name the template `DynamicArray`, with the data type as a template variable (as in the lecture topic 4 notes). Write the public interface exactly the same as StaticArray, so that they are interchangeable in any application.

You decide upon the default initial capacity to use in the template.

Write a test driver named **DynamicArrayDriver.cpp**, testing all functions, ifndef, object copy, and object assignment. Test its expandability by using an index value greater than the capacity, and see what it does to size and capacity. Test its constructor's default parameter by creating and testing two DynamicArray objects -- one with no parameter, defaulting to the default initial capacity, and one with an initial capacity set in the test driver.

Submit the two files (1 CPP and 1 H) to the class website for credit.

LAB 4b: Write A Dynamic Array Application `[ MyDynamicArray.cpp ]`
Purpose. The purpose of this lab is for you to apply your own templated class. If it was fully tested in 4a above, then it should work fine in this application.

Write **`MyDynamicArray.cpp`**, using your**DynamicArray.h** template. Modify your lab 3b app to use your new DynamicArray instead of StaticArray. As in 3b, make the initial capacity 100, either as the default initial capacity in the template, or as in the DynamicArray declaration in the app. No other changes should be made, but now the app should allow index values greater than 100 to be entered, like this (user input in blue):

```
Input an index and a value [Q to quit]: 33 12
Input an index and a value [Q to quit]: 1000 100
Input an index and a value [Q to quit]: -1 234
Input an index and a value [Q to quit]: q

I stored this many values: 2
The values are:
  33 120
  1000 100

Input an index for me to look up [Q to quit]: 33
Found it -- the value stored at 33 is 120
Input an index for me to look up [Q to quit]: 1000
```

```
Found it -- the value stored at 1000 is 100
Input an index for me to look up [Q to quit]: Q
```

How to pause a console program: *show / hide*

GRADING RUBRIC *show/hide*
